

発表概要

Julia 言語への Actor の導入

中田 秀基^{1,a)}

2021年7月20日発表

Julia 言語はインタラクティブな言語でありながら、LLVM を利用した JIT により高速な実行が可能なことから科学技術計算向け言語として注目を集めている。Julia は分散環境での実行をデフォルトでサポートしており、リモートノードでの関数実行、リモートチャンネルへの書き込み、Future による同期などが利用できる。しかし、リモート変数への参照を持つことができないため、リモートノードに状態を保持しておき、その状態を利用した計算を行うことが難しい。本発表では Julia 言語に Actor を導入することで、分散並列計算の実行を容易にすることを試みた。Actor は 1980 年代に提唱された並列計算モデルに基づく実行モデルで 1990 年代に広く研究されたが、近年 Akka が採用するなど再評価されている。Actor は状態を保持したシングルスレッドでメッセージを処理するオブジェクトとしてとらえることができ、粗粒度の並列実行モデルとして適している。本発表ではユーザに提供する Actor API のモデルに付いて議論し、Julia 既存の言語機能を利用することで効率的に Actor を実現できることを示す。また、本実装のマイクロベンチマークでの性能評価結果を示す。

Presentation Abstract

An Implementation of Actor in Julia Language

HIDEMOTO NAKADA^{1,a)}

Presented: July 20, 2021

Julia language is one of the most promising programming languages for the next generation scientific programming language because of its fast execution thanks to LLVM based JIT compilation. Julia supports distributed computing with remote function invocation, remote channel communication, and Future based synchronization, out of the shelf. Unfortunately, it is not easy to write programs that utilize stateful objects on the remote node, since Julia does not support remote object reference. In this presentation, we introduce Actor to Julia. Actor is a parallel computation model that is proposed in the '80s and thoroughly investigated in the '90s. Recently, Actor is re-gathering attentions as a parallel processing model and featured in several libraries, including Akka, for Java and Scala. We discuss the API design and implementation of Actor in Julia, and show that we can easily implement it with the existing language construct of Julia. We also show the performance of the implementation with microbenchmarks.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

¹ 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology, Tsukuba, Ibaraki 305-8560, Japan

^{a)} hide-nakada@aist.go.jp